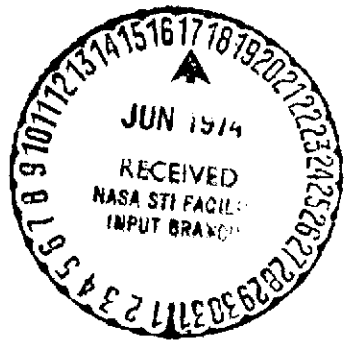
NASA TECHNICAL NOTE

NASA TN D-7580

NASA TN D-7580



AN ALGORITHM FOR A GENERAL CLASS OF ROUTING PROBLEMS DERIVED FROM HUYGENS' PRINCIPLE

by Lee M. Avis and George R. Young

Langley Research Center
Hampton, Va. 23665

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • JUNE 1974

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TN D-7580		

4. Title and Subtitle	5. Report Date
AN ALGORITHM FOR A GENERAL CLASS OF ROUTING PROBLEMS DERIVED FROM HUYGENS' PRINCIPLE	June 1974
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Lee M. Avis and George R. Young	L-9342
	10. Work Unit No.
9. Performing Organization Name and Address	790-93-08-01
NASA Langley Research Center	11. Contract or Grant No.
Hampton, Va. 23665	
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Technical Note
National Aeronautics and Space Administration	14. Sponsoring Agency Code
Washington, D.C. 20546	

15. Supplementary Notes

16. Abstract

If a set of N points or nodes with a nonnegative cost associated with each ordered pair is known, it is desired to find a path from one given node to another given node which minimizes the cost sum. An algorithm is presented which yields a global minimum solution after at most N - 1 iterations or on a typical large third-generation computer, after 1 hour of computation time for a 10 000-node problem. The rapid-access data storage capacity demanded by the algorithm is approximately 3N words for costs read in from slow-access storage or 2N words for calculable costs. The time-storage requirements of the algorithm compare favorably with those of dynamic programing or any other optimal path algorithm known to the authors. When the problem is viewed as a discretized optimal control problem, after N - 1 iterations, an optimal control or node transition is established for each of the N nodes or states; thus, the algorithm can be applied to situations where there may be errors in the control that necessitate a closed loop control philosophy.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Routing problems Optimal control Operations research Dynamic programing Algorithms Numerical optimization	Unclassified – Unlimited STAR Category 30

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	19	$3.00

*For sale by the National Technical Information Service, Springfield, Virginia 22151

AN ALGORITHM FOR A GENERAL CLASS OF ROUTING PROBLEMS DERIVED FROM HUYGENS' PRINCIPLE

By Lee M. Avis and George R. Young
Langley Research Center

SUMMARY

If a set of N points or nodes with a nonnegative cost associated with each ordered pair is known, it is desired to find a path from one given node to another given node which minimizes the cost sum. An algorithm is presented which yields a global minimum solution after at most $N - 1$ iterations or on a typical large third-generation computer, after 1 hour of computation time for a 10 000-node problem. The rapid-access data storage capacity demanded by the algorithm is approximately 3N words for costs read in from slow-access storage or 2N words for calculable costs. The time-storage requirements of the algorithm compare favorably with those of dynamic programing or any other optimal path algorithm known to the authors. When the problem is viewed as a discretized optimal control problem, after $N - 1$ iterations, an optimal control or node transition is established for each of the N nodes or states; thus, the algorithm can be applied to situations where there may be errors in the control that necessitate a closed loop control philosophy.

INTRODUCTION

The following class of routing problems is considered: If a network of nodes with a nonnegative cost associated with each ordered pair of nodes is given, it is desired to find a minimum cost path from one given node to another given node.

A method of finding global optimum paths through a network of nodes is presented as a discretized algorithmic form of the Huygens' Principle (ref. 1) which can be stated as

> An optical wave front propagates through space or matter as if each point on the front were a source of a new wave front and the envelope of these new (secondary) wave fronts in advance of the given (primary) front were a subsequent primary wave front. The line joining a secondary source and the envelope of secondary wave fronts in the shortest optical distance approximates a primary optical ray; in general, the nearer the envelope to the secondary source, the better the approximation.

The algorithm presented herein constructs (exact) optimal paths through the network of nodes as the Huygens' Principle constructs light rays through space or matter, where optical distances and propagating wave fronts are replaced by, respectively, cost distances and propagating cost contours.

The algorithm is presented and developed as a means of solving the fixed end points optimal path problem stated in the section "Statement of the Problem"; extensions to variable ends points and optimal control problems are discussed in the section "Applications to Optimal Control Problems With Multinode Goals and Origins."

The Huygens' Principle foundation of the current algorithm should lead to much further generalizations and extensions of the technique developed here than that of a similar algorithm developed by E. W. Dijkstra (ref. 2).

SYMBOLS

a,b computer times in microseconds for certain operations

$c(I_0, \ldots, j)$ cost of path $(I_0, \ldots, j)$

$(I_0, \ldots, j)$ path from node I_0 to node j

$(I_0 \rightarrow j)$ optimal path from node I_0 to node j

(i,j) direct path from node i to node j

I_0, I_f starting node and goal node, respectively, defining the end points of the optimal path

I_s starting node defined on page 9

J_k $(k + 1)$th node to be occupied; defined on page 10

j_k unoccupied node for which $c(I_0, \ldots, j)$ is a minimum at kth step

K_{ij} cost matrix element; the cost of the direct path from node i to node j

N number of nodes in problem

n_1, n_2 indices of refraction

q any node on path of equation (2)

r,s,t,j running indices

$\{S_k\}$ $(k+1)$th set of occupied nodes, $\{J_0,J_1,J_2,\ldots,J_k\}$

T_k cost of an optimal path from node I_0 (or J_0) to node J_k

ΔT $T_{k+1} - T_k$ at the $(k+1)$th iteration of algorithm, which is obtained operationally as minimum of τ values

u_i next to last node of optimal path from node I_0 to node i for any occupied node i; for an unoccupied node i, u_i is a current estimate of the next to last node of optimal path from node I_0 to node i

$\sigma^{(k)}_{J_r,j}$ at $(k+1)$th iteration, the cost distance remaining to reach (occupy) node j along the direct path (J_r,j) where J_r is an occupied node (defined after eq. (4))

$\tau^{(k)}_j$ at $(k+1)$th iteration, the minimum over all occupied nodes J_r of $\sigma^{(k)}_{J_r,j}$ for node j unoccupied; for node j occupied, $\tau^{(k)}_j = \infty$

τ'_j variable of algorithm in step (6) on page 7

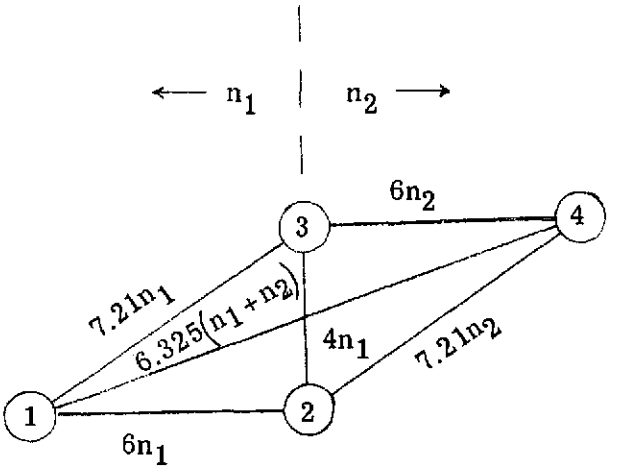
STATEMENT OF THE PROBLEM

A set of N nodes, numbered arbitrarily from 1 to N, together with a cost matrix K_{ij} is known where $i,j = 1,\ldots,N$ and $K_{ij} \geq 0$; find a sequence of nodes $(I_0,I_1,I_2,\ldots,I_f)$ which minimizes $\sum_{i=0}^{f-1} \left[K_{I_i,I_{i+1}}\right]$ when I_0 and I_f are known but not the number of nodes in the path $f+1$. That is, a minimum cost path from I_0 to I_f is sought. A sequence of nodes is called the "path" through those nodes and the cost matrix defines the cost of all paths.

EXAMPLE PROBLEM

A simple four-node example problem is solved to illustrate the principles of the algorithm and its kinship to Huygens' Principle. Consider the four nodes at the vertices

of the parallelogram of sketch (a). The costs associated with the sides and diagonals of the parallelogram are indicated in sketch (a). Suppose an optimal path from node 1 to node 4 is desired. The cost per unit path length is n_2 to the right of (but excluding) the diagonal (3,2) and n_1 elsewhere. If n_1 and n_2 are interpreted as indices of refraction, then the light ray path from node 1 to node 4 of minimum optical path length (or equivalently, minimum travel time) is sought when the ray is confined to the sides and diagonals of the parallelogram. A discretized simulation of the refraction of light rays passing from one medium to another of different refractive index has been defined.



Sketch (a).- Four-node refraction problem.

Let $n_1 = 1$ and $n_2 = 3$. The matrix of costs K_{ij} associated with the ordered node pairs or links is then

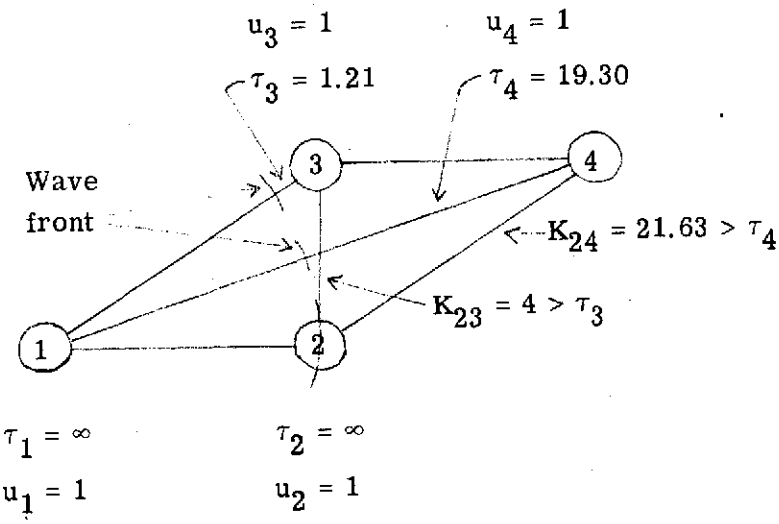
$$[K] = \begin{bmatrix} 0 & 6 & 7.21 & 25.30 \\ 6 & 0 & 4 & 21.63 \\ 7.21 & 4 & 0 & 18 \\ 25.30 & 21.63 & 18 & 0 \end{bmatrix}$$

Consider a wave front originating at node 1 and propagating at a speed inversely proportional to the cost per unit link length; that is, the wave front is a contour of constant cost. In order to indicate that node 1 has been encountered by the wave front or "occupied," τ_1 is set equal to a very large number (written ∞). The shortest cost distances to the unoccupied nodes 2, 3, and 4 from the wave front along direct links to these nodes are specified by, respectively, $\tau_2 = 6$, $\tau_3 = 7.21$, and $\tau_4 = 25.30$. To complete the description of the wave front, record the direct link associated with each τ_j by $u_j = i$ where j is the node and the direct link is (i,j). Thus, at the beginning, $u_1 = u_2 = u_3 = u_4 = 1$.

The next node reached (occupied) by the wave front is node 2, since τ_2 is the minimum τ. In reaching node 2, the wave front has advanced τ_2 or six cost units, and so the τ values of the unoccupied nodes are decremented by 6. Then τ_2 is set equal to ∞ to flag node 2 as occupied. Thus the τ values become

$$\tau_1 = \infty \qquad \tau_2 = \infty \qquad \tau_3 = 1.21 \qquad \tau_4 = 19.30$$

The newly occupied node 2 now becomes the source of a secondary wave front that is consistent with Huygens' Principle. As shown in sketch (b) and by the K matrix, the wave front originating at node 2 (now a single point at node 2) is more cost distant from the unoccupied nodes 3 and 4 than is the wave front originating at node 1; therefore, τ_4 is not changed to K_{24} nor is u_4 changed to 2, and τ_3 is not changed to K_{23} nor is u_3 changed to 2.



Sketch (b).- Node 2 is occupied.

The possibility that the optimal path from node 1 to any unoccupied node passes through node 2 has thus been eliminated, and the secondary wave originating at node 2 can be neglected. The Huygens' Principle has a provision similar to that exercised here: that all portions of the secondary waves which are behind the advancing wave envelope are neglected.

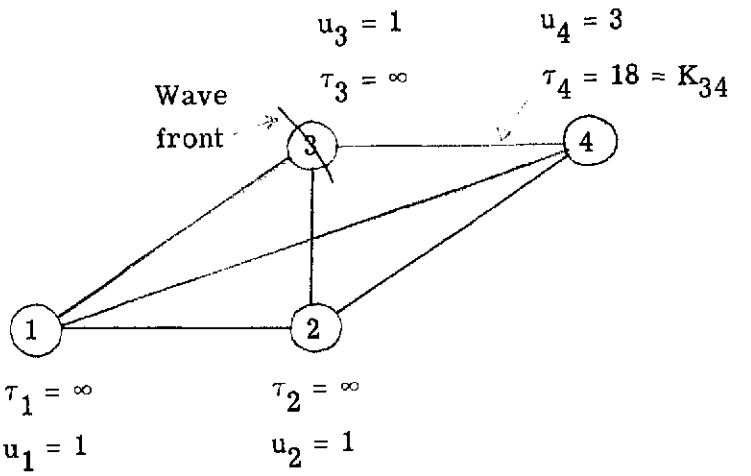
At this stage, τ_3 is the minimum τ; thus, node 3 is the next node reached by the (composite) wave front. By proceeding as before, the new τ_4 is set equal to $\tau_4 - \tau_3 = 19.30 - 1.21 = 18.09$ and then τ_3 is set equal to ∞. Now node 3 becomes a new wave source. Since this third wave is slightly "closer" to node 4 than is the old composite wave, τ_4 is changed from 18.09 to $K_{34} = 18$ and u_4 is changed from 1 to 3. See

sketch (c). Finally, τ_4 being the minimum τ, node 4 is the next node to be reached by the composite wave. An optimal path from node 1 to node 4 is defined by

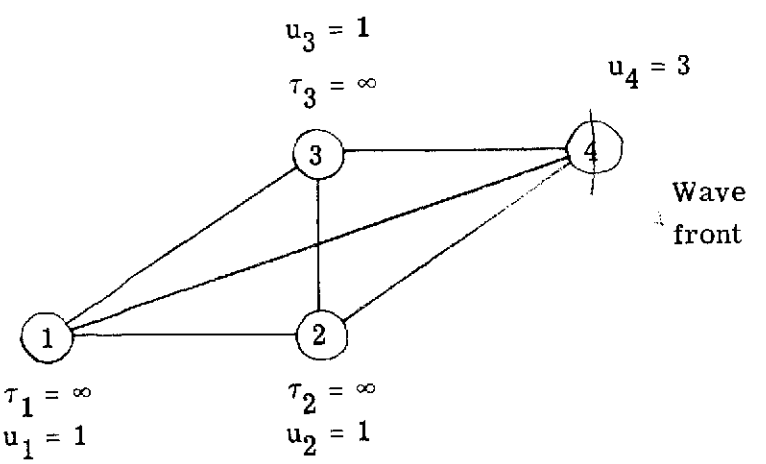
$$u_4 = 3$$

$$u_3 = 1$$

or path (1,3,4). See sketch (d). The optimal paths from node 1 to node 2 and to node 3 have also been found; $u_2 = 1$ means that path (1,2) is optimal (once τ_2 becomes ∞) and $u_3 = 1$ means that path (1,3) is optimal. The u values define an optimal transition to each occupied node for a given starting node. If the problem is worked in reverse from node 4 to node 1 (reversing the indices of K if K is asymmetric), the u values would define an optimal transition from each occupied node for a given goal node.



Sketch (c).- Node 3 is occupied.



Sketch (d).- Node 4 is reached.

The Huygens' analogue to the u values are those vectors (rays) connecting each secondary source to the envelope of secondary waves in the shortest optical distance. The algorithm replaces optical distance with cost distance and replaces the rays with pointers to the sources, that is, the u values. The envelope of secondary waves is analogous to the composite wave or the cost contour of the algorithm. A distinction between Huygens' Principle and the algorithm is that light, according to Huygens' Principle, propagates so that obstructions cast shadows, whereas the cost contour constructed by the algorithm spreads throughout all available "space." (The analogue to an obstruction is a set of nodes which can be reached from nodes outside the set only through links of very high cost.)

DESCRIPTION OF ALGORITHM

The algorithm is now presented in a form convenient for calculation. The elements of K, the cost matrix, as defined previously, can either be calculated as needed or precomputed and stored in auxiliary storage and read in as needed. Let I_0 be the starting node and I_f be the goal node and N be the number of nodes. The algorithm follows.

Step	Operation	Comment
(1)	$\tau_j = K_{I_0,j}$; $u_j = I_0$ $(j = 1,\ldots,N)$	Initialization of iteration loop.
(2)	$\tau_{I_0} = \infty$	Flags I_0 as occupied.
(3)	$\Delta T \left(= \tau_{j_0}\right) = \min\left\{\tau_j\right\}$ $(j = 1,\ldots,N)$	Begin iteration loop.
(4)	If $j_0 = I_f$, go to step (10)	Stop iteration when goal node I_f is occupied.
(5)	$\tau_{j_0} = \infty$	Flags j_0 as occupied.
(6)	$\tau'_j = K_{j_0,j}$ $(j = 1,\ldots,N)$	If K elements are not calculated, read j_0th row of K into τ'. Skip this operation if K elements are calculated.
(7)	$\tau_j = \tau_j - \Delta T$ $\left(\text{All } \tau_j \neq \infty\right)$	
(8)	$\tau_j = \min\left\{\tau_j, \tau'_j\right\}$ $\left(\text{All } \tau_j \neq \infty\right)$	If K elements are calculated, the τ'_j array is not needed and can be replaced by $K_{j_0,j}$ to reduce data storage by N words.
(9)	$u_j = j_0$ $\left(\text{for all } j \text{ having } \tau_j = \tau'_j\right)$. Go to step (3).	
(10)	Output j_0; $j = j_0$; Cost = 0.0	Initialization of output loop.
(11)	$i = u_j$	Output loop (steps (11) to (15)). Yields an optimal path from I_0 to I_f (in reverse order), the total path cost, and the incremental path cost.
(12)	Cost = Cost + K_{ij}	
(13)	Output i, cost	
(14)	If $i = I_0$, exit	
(15)	$j = i$. Go to step (11).	

APPLICATIONS TO OPTIMAL CONTROL PROBLEMS WITH MULTINODE GOALS AND ORIGINS

The optimal path problem is cast in the form of an optimal control problem by identifying the nodes as states of a system which is to be brought from an initial state to a specified final state through a sequence of states minimizing the sum of the state transition costs. Each state transition cost is assumed to be a function only of that state transition; this assumption amounts to a definition of the concept of "states," in that the cost matrix defines the system. The states or nodes might represent a discretized approximation to a continuum state space. Disallowed state transitions are assessed a high cost level to prevent, or at least flag in the output, the "forbidden" transitions. The "forbidden" flag is set less than the "occupied node" flag (∞) to avoid premature assignment of nodes to the occupied status.

As pointed out previously, if the optimal path problem is worked in reverse (with the goal node as the initially occupied node) with a transposed K matrix, then the algorithm establishes an optimal transition u_j from each node j which becomes occupied. The u values thus define an optimal control, or an optimal way to proceed, from each occupied node, when a specified goal node is given. Since one additional node is occupied upon each iteration, $N - 1$ iterations are required to generate the complete field of optimal controls over the state space. The complete field of optimal controls defines the optimal procedure even after a control error, which allows closed loop control with state information feedback.

After $N - 1$ iterations, all nodes are occupied, including those "isolated" nodes, if any, from which the goal node can be reached only by a series of transitions including at least one forbidden transition. In this situation, the u values contain some forbidden transitions, which, however, are flagged in the output by the associated high cost. In some cases, it is convenient and computationally advantageous to remove isolated nodes from the state space beforehand.

The modifications to the algorithm for generating the complete optimal control field are

(1) Replace the K matrix with its transpose $\left[K^T\right]$

(2) Wherever I_0 occurs, replace it with I_f, the goal node

(3) Initialize an iteration number $k = 0$ before the third step of the algorithm

(4) Replace the test at step (4) by an operation incrementing k by 1 and then an operation testing k for the value $N - 1$, a successful test triggering a shift to the output

(5) Replace the output loop (steps (10) to (15)) by a routine which outputs the u_j array and the associated costs $K^T_{u_j,j}$

Once the complete optimal control field is generated, the optimal path(s) from any node(s) to the goal node I_f can be output as the sequence(s) of optimal controls beginning at the desired starting node(s) I_S; the sequence(s) are of the form $\left(I_S, u_{I_S}, u_{u_{I_S}}, \ldots, I_f\right)$.

If only the optimal paths from each of a set of starting nodes to a goal node and the associated costs are desired (for example, all optimal paths originating within a neighborhood of a nominal starting node), then $N - 1$ iterations of the algorithm, in general, are not necessary. The modifications to the algorithm to allow less than $N - 1$ iterations are as follows:

Steps (1) to (3) are the same as those for complete optimal control field

Step (4) Replace the test at step (4) by an operation incrementing k by 1 when the newly occupied node j_0 is one of the desired starting nodes and perform an operation testing k for the number (quantity) of desired starting nodes, and then switch to output when k equals the number of starting nodes

Step (5) Replace the output loop, steps (10) to (15), by a routine which outputs the sequences of optimal controls $\left(I_S, u_{I_S}, u_{u_{I_S}}, \ldots, I_f\right)$ for each desired starting node I_S and the associated costs

A generalization of the problem of path optimization with fixed end points, optimization over all paths with starting and goal nodes each in specified respective subsets of the set of N nodes, is solvable in one application of the algorithm. A simple procedure is to set all $K_{ij} = 0$ for both i and j in the goal subset, transpose the K matrix, set I_0 equal to any node in the goal subset, change the stopping rule test (step (4)) from "$j_0 = I_f$?" to "$j_0 \in \{\text{starting subset}\}$?", and change the exit rule test in the output loop (step (14)) from "$i = I_0$?" to "$i \in \{\text{goal subset}\}$?".

THEORETICAL BASIS OF ALGORITHM

In this section, theorems are proven which provide the basis for the algorithm.

As demonstrated by the example problem (and shown in the following), the algorithm places each node in the occupied status, in sequence, in increasing order of the cost of optimal paths from the starting node I_0 to each of the nodes. It is first reasoned that closed loops need not be considered in seeking an optimal path; thus, only a finite number of paths need to be considered. From this finite number of path costs, a unique minimum

can be selected. It is proven that there is an optimal path from I_0 to an unoccupied node j such that (a) all nodes in the path except j are occupied and (b) the cost of the path is not less than the cost of any optimal path from I_0 to any occupied node and not greater than the cost of any path from I_0 to any unoccupied node. The procedure for computing the cost distance to node j is then demonstrated. This procedure, applied recursively, yields a global optimum path.

Assume that a ranking (with ties permitted) exists for the costs of all paths originating at I_0. The existence of a unique minimum path cost over all paths from I_0 to any given node is shown to result from the nonnegativity of the elements of the cost matrix K by the following statement.

For any path from I_0 to a given node j containing any node q more than once (that is having a closed loop), $(I_0,n_1,n_2,\ldots,q,m_1,m_2,\ldots,q,\ldots,j)$, there is a path from I_0 to j of no greater cost obtained by deleting from the node sequence all nodes following the first occurrence of q up to and including the last occurrence of q; for the nonnegativity of the cost elements implies that the cost of the sequence of nodes deleted is nonnegative. The number of paths from I_0 to j with no multiply occurring nodes is finite and, by assumption, their costs can be ranked; therefore, a path from I_0 to j of unique minimum cost over all paths from I_0 to j exists, this being the minimum over the paths from I_0 to j with no multiply occurring nodes.

Let the minima over the costs of all paths from I_0 to each of the N nodes be sequenced in a monotonically nondecreasing order: $T_0,T_1,T_2,\ldots,T_{N-1}$. For each T_k, associate a node J_k so that the cost of an optimal path from I_0 to J_k is T_k for $k = 0,1,2,\ldots,N-1$. (The sequence of J_k values is the sequence in which each node is placed in the occupied status. This sequence is not necessarily unique because of the possibility of ties, which are arbitrarily ordered.) Let $\{S_k\}$ be the set of all J_r for $r = 0,1,2,\ldots,k$; $\{S_k\}$ is to be identified as the $(k+1)$th set of occupied nodes. For simplicity, the condition that $J_0 = I_0$ is chosen without loss of generality.

The following theorem demonstrates the constructive nature of the algorithm.

> Theorem: There is an optimal path from I_0 to some $j \notin \{S_k\}$ of cost, T_{k+1}, such that all nodes in the path except j belong to $\{S_k\}$, for $k = 0,1,2,\ldots,N-2$.

Proof: By definition of $\{S_k\}$, T_k, and J_r for $r = 0,1,\ldots,k$,

$$T_{k+1} = \min_{j \notin \{S_k\}} \{c(I_0 - j)\} \tag{1}$$

where $(I_0 - j)$ is an optimal path from I_0 to j and $c(I_0 - j)$ is the associated cost. Let j_k be a j for which $c(I_0 - j)$ is minimized. Then,

$$T_{k+1} = c(I_0 - j_k) \tag{2}$$

Let q be any node on the path of equation (2). Then,

$$T_{k+1} = c(I_0, \ldots, q, \ldots, j_k)$$

If $(I_0, \ldots, q)$ or $(q, \ldots, j_k)$ is not optimal, then $(I_0, \ldots, q, \ldots, j_k)$ is not optimal. By contradiction,

$$T_{k+1} = c(I_0 - q - j_k)$$

By the nonnegativity of K_{ij},

$$c(I_0 - q) \leqq c(I_0 - q - j_k) = T_{k+1} \tag{3}$$

Suppose $q \notin \{S_k\}$. Then q is a candidate j in equation (1) and

$$c(I_0 - q) \geqq T_{k+1}$$

Therefore by equation (3)

$$c(I_0 - q) = T_{k+1}$$

(Note the incidental result that $c(q - j_k) = 0$.)

The algorithm determines the path-defining u values indirectly by means of recursive minimization over the incremental costs (the τ values) of completing optimal paths to unoccupied nodes. The essential features of the optimal path determination through recursive minimization are set forth in the following theorem; the corollary to the theorem allows simplification of the computation and reduction of data storage. $\left(T_{k+1} - T_k \text{ is the } \Delta T \text{ of the algorithm at the } (k+1)\text{th iteration.}\right)$

Theorem:

$$T_{k+1} - T_k = \min_{\substack{r \in \{0,1,\ldots,k\} \\ j \notin \{S_k\}}} \left\{ \sigma^{(k)}_{J_r, j} \right\} \qquad \left(k \in \{0, \ldots, (N-2)\}\right) \tag{4}$$

where

$$\sigma^{(k)}_{J_r,j} = \sigma^{(k-1)}_{J_r,j} - \left(T_k - T_{k-1}\right) \qquad \left(r \in \{0,1, \ldots,(k-1)\};\ j \notin \{S_k\}\right)$$

$$\sigma^{(k)}_{J_r,j} = K_{J_r,j} \qquad \left(r = k;\ j \notin \{S_k\}\right)$$

Furthermore, there is an optimal path from I_0 to J_{k+1} of the form $\left(I_0 - J_R, J_{k+1}\right)$ where J_R and J_{k+1} are the J_r and j, respectively, which minimize $\sigma^{(k)}_{J_r,j}$.

Proof: Equation (1) in the proof of the first theorem is

$$T_{k+1} = \min_{j \notin \{S_k\}} \{c(I_0 - j)\}$$

which, by the first theorem and the definition of the J_r, can be written as

$$T_{k+1} = \min_{\substack{r \in \{0,1, \ldots,k\} \\ j \notin \{S_k\}}} \{c(I_0 - J_r,j)\} \tag{5}$$

By definition,

$$c(I_0 - J_r) \equiv T_r$$

Thus, equation (5) can be written as

$$T_{k+1} = \min_{\substack{r \in \{0,1, \ldots,k\} \\ j \notin \{S_k\}}} \{T_r + c(J_r,j)\}$$

By subtracting T_k and substituting $K_{J_r,j}$ for $c(J_r,j)$,

$$T_{k+1} - T_k = \min_{\substack{r \in \{0,1,\ldots,k\} \\ j \notin \{S_k\}}} \left\{ K_{J_r,j} - \left(T_k - T_r\right) \right\} \tag{6}$$

From the definition of $\sigma^{(k)}_{J_r,j}$, it follows that

$$\sigma^{(k)}_{J_r,j} = \sigma^{(r)}_{J_r,j} - \left(T_k - T_r\right) = K_{J_r,j} - \left(T_k - T_r\right) \qquad \left(r \in \{0,1,\ldots,k\};\ j \notin \{S_k\}\right)$$

Therefore, equation (6) can be written as

$$T_{k+1} - T_k = \min_{\substack{r \in \{0,1,\ldots,k\} \\ j \notin \{S_k\}}} \left\{ \sigma^{(k)}_{J_r,j} \right\}$$

which establishes the first part of the theorem (eq. (4).) Let J_R and J' be the J_r and j, respectively, which minimize $\sigma^{(k)}_{J_r,j}$ in equation (4). Then, by equations (5) and (4),

$$T_{k+1} = c(I_0 \rightarrow J_R, J')$$

By equation (1), the path $(I_0 \rightarrow J_R, J')$ is optimal; thus, by definition of the J_r, one can set $J_{k+1} = J'$. Therefore, $(I_0 \rightarrow J_R, J_{k+1})$ is optimal.

Corollary:

$$T_{k+1} - T_k = \min_{j \notin \{S_k\}} \left\{ \tau^{(k)}_j \right\} \qquad \left(k \in \{0,\ldots,(N-2)\}\right)$$

where $\tau_j^{(k)}$ is defined by

$$\tau_j^{(k)} \equiv \min\left\{\left[\tau_j^{(k-1)} - \left(T_k - T_{k-1}\right)\right], \left[K_{J_k,j}\right]\right\} \qquad \left(k > 0; \quad j \notin \{S_k\}\right)$$

$$\tau_j^{(0)} \equiv K_{J_0,j} \qquad \left(j \neq J_0\right)$$

Proof: The proof is inductive on k. Suppose $\tau_j^{(k)} = \min_{t \in \{0,1,\ldots,k\}} \left\{\sigma_{J_t,j}^{(k)}\right\}$ (all $j \notin \{S_k\}$ and some $k \in \{0,1,\ldots,(N-3)\}$). Then, by equation (4) and the hypothesis,

$$T_{k+1} - T_k = \min_{j \notin \{S_k\}} \left\{\tau_j^{(k)}\right\}$$

By equation (4),

$$\sigma_{J_r,j}^{(k+1)} = \sigma_{J_r,j}^{(k)} - \left(T_{k+1} - T_k\right) \qquad \left(r \in \{0,1,\ldots,k\}; \quad j \notin \{S_{k+1}\}\right)$$

$$\sigma_{J_r,j}^{(k+1)} = K_{J_r,j} \qquad \left(r = k+1; \quad j \notin \{S_{k+1}\}\right)$$

Thus,

$$\min_{r \in \{0,1,\ldots,(k+1)\}} \left\{\sigma_{J_r,j}^{(k+1)}\right\} = \min\left\{\min_t \left\{\sigma_{J_t,j}^{(k)} - \left(T_{k+1} - T_k\right)\right\}, K_{J_{k+1},j}\right\}$$

where $t \in \{0,1,\ldots,k\}$, $j \notin \{S_{k+1}\}$ or by the hypothesis

$$\min_{r \in \{0,1,\ldots,(k+1)\}} \left\{\sigma_{J_r,j}^{(k+1)}\right\} = \min\left\{\left[\tau_j^{(k)} - \left(T_{k+1} - T_k\right)\right], \left[K_{J_{k+1},j}\right]\right\} \equiv \tau_j^{(k+1)}$$

Since

$$\tau_j^{(k)} = \min_{t\in\{0,1,\ldots,k\}} \left\{\sigma^{(k)}_{J_t,j}\right\} \rightarrow \begin{cases} T_{k+1} - T_k = \min\limits_{j\notin\{S_k\}} \left\{\tau_j^{(k)}\right\} \\ \tau_j^{(k+1)} = \min\limits_{r\in\{0,1,\ldots,(k+1)\}} \left\{\sigma^{(k+1)}_{J_r,j}\right\} \end{cases}$$

for all $j \notin \{S_k\}$ and for any $k \in \{0,1,\ldots,(N-3)\}$, in order to complete the inductive proof, it will be shown that for $k = 0$,

$$\tau_j^{(k)} = \min_{t\in\{0,\ldots,k\}} \left\{\sigma^{(k)}_{J_t,j}\right\} \qquad \left(\text{All } j \notin \{S_k\}\right)$$

For $k = 0$ by definition,

$$\tau_j^{(k)} = K_{J_0,j} \qquad \left(\text{All } j \notin \{S_k\}\right)$$

$$\min_{t\in\{0,\ldots,k\}} \left\{\sigma^{(k)}_{J_t,j}\right\} = \sigma^{(0)}_{J_0,j} \equiv K_{J_0,j}$$

Therefore,

$$\tau_j^{(k)} = \min_{t\in\{0,\ldots,k\}} \left\{\sigma^{(k)}_{J_t,j}\right\}$$

Note that the algorithm defines the optimal paths by updating the u_j when the τ_j are updated by

$$\tau_j^{(k)} = \min\left\{\left[\tau_j^{(k-1)} - \left(T_k - T_{k-1}\right)\right], \left[K_{J_k,j}\right]\right\}$$

whereupon if $K_{J_k,j} < \tau_j^{(k-1)} - \left(T_k - T_{k-1}\right)$, then u_j is set to J_k. Since the proof of the corollary shows that

$$\tau_j^{(k)} = \min_{t \in \{0, \ldots, k\}} \left\{\sigma_{J_t,j}^{(k)}\right\} \qquad \left(k \in \{0, \ldots, (N-2)\};\ j \notin \{S_k\}\right)$$

then, when the minimum $\tau_j^{(k)}$ is found, u_j is the J_t which minimizes $\sigma_{J_t,j}^{(k)}$. Thus, by the second theorem, there is an optimal path from I_0 to J_{k+1} of the form $\left(I_0 - u_{J_{k+1}}, J_{k+1}\right)$ where J_{k+1} is the j which minimizes $\tau_j^{(k)}$.

COMPUTATIONAL CONSIDERATIONS

The rapid-access data storage demanded by the algorithm is approximately 3N words (for the u_j, τ_j, and τ_j') for cost elements read in from slow-access storage. (The rapid-access data storage can be reduced by up to N words by reading at step (6) partial rows of the K matrix into an abbreviated τ' array, working with this array down through step (9) with appropriate changes to running index limits, and cycling back through step (6) for the next τ' array until the j_0th row of K is exhausted. The additional computation time that would result might be acceptable if core storage were a serious constraint.) For cost elements calculated as needed, the rapid-access data storage is approximately 2N (for the u_j and the τ_j). Now consider computation time, or more precisely, residence time in core memory.

The numbers of operations performed for the worst case of N - 1 iterations are as follows:

Operation	Total times performed	Breakdown
Comparisons	$2N^2$	Step (7), N^2 Step (8), $1/2\ N^2$ Step (3), $1/2\ N^2$
Additions or subtractions	$1/2\ N^2$	Step (7), $1/2\ N^2$
Substitutions	$2N^2$	Step (3), N^2 Step (8), $1/2\ N^2$ Step (9), $1/2\ N^2$

Operation	Total times performed	Breakdown
Calculations of τ or	$1/2\ N^2$	Step (8), $1/2\ N^2$
Row read-in of τ,τ' from auxiliary storage	2N	Step (6), 2N

For those particular operations where an exact count of frequency could not be made, the worst case frequency is taken.

For large third-generation machines, the computation times of each of the listed operations except the last two are in the approximate range 1 to 10 microseconds. The time used in a row read-in of τ,τ' or a read-in of a single element is of the order of 100 milliseconds for disk storage systems. Thus, for costs read-in rather than calculated, the computation time is approximately

$$\text{Time} \lessapprox 4.5 \times 10^{-6}\ aN^2 + 0.2N \text{ seconds}$$

where a is a characteristic computation time in microseconds for these listed operations (excluding the last two), and $1 \lessapprox a \lessapprox 10$. For the case of computed costs, the computation time is approximately

$$\text{Time} \lessapprox (4.5a + b/2) \times 10^{-6}\ N^2 \text{ seconds}$$

where b is the computation time in microseconds for one τ computation.

COMPARISON WITH OTHER ALGORITHMS

The data core storage and maximum numbers of operations for the proposed algorithm, a dynamic programing algorithm (ref. 3), and an algorithm due to Ford and Fulkerson (ref. 4) are presented in the following table. The term "elementary operations" refers to replacement, addition, subtraction, multiplication, division, or simple one-branch or two-branch logical operations.

The Ford and Fulkerson algorithm is limited in practice to problems with no more than a few hundred nodes because of the rapid increase in the number of operations with increasing numbers of nodes. However, this algorithm works under a less restrictive assumption than that of nonnegative costs; the cost of any closed loop is nonnegative. It can be seen from the table that the proposed algorithm is comparable in storage to the Ford and Fulkerson algorithm but requires less computation time by a factor of the order of N^2. It can also be seen from the table that the general routing problem with nonnegative costs is solvable by the proposed algorithm in, for worst cases, less computer time

by nearly a factor of N and with somewhat less core storage than the dynamic programing algorithm. This advantage can be exploited to solve general routing problems with an order of magnitude more nodes than those solvable by dynamic programing in the same run time.

Algorithm	Source of cost elements	Data core storage	Number of elementary operations	Number of cost element computations	Number of accesses of auxiliary storage
Dynamic programing	Computed	$\approx 3N$	$\approx 3N^3$	$\approx N^3$	-------
Dynamic programing	Read in from auxiliary storage	$\approx 4N$	$\approx 3N^3$	-------	$\approx N^2$
Dynamic programing	Stored in core	$\approx N^2$	$\approx 3N^3$	-------	-------
Ford and Fulkerson	Computed	$\approx 2N$	$\approx 1/8\ N^4$	$\approx 1/8\ N^4$	-------
Ford and Fulkerson	Read in from auxiliary storage	$\approx 3N$	$\approx 1/8\ N^4$	-------	$\approx 1/8\ N^3$
Ford and Fulkerson	Stored in core	$\approx N^2$	$\approx 1/8\ N^4$	-------	-------
Proposed	Computed	$\approx 2N$	$\approx 9/2\ N^2$	$\approx 1/2\ N^2$	-------
Proposed	Read in from auxiliary storage	$\approx 3N$	$\approx 9/2\ N^2$	-------	$\approx 2N$

CONCLUDING REMARKS

Huygens' Principle has been used as the motivation to develop an algorithm for solving a general class of routing problems. If a nonnegative cost for each ordered pair of a set of N nodes is known, the algorithm can find the path of minimum cost sum from one given node to another given node in at most N - 1 iterations. By viewing the problem as a discretized optimal control problem, after N - 1 iterations, an optimal control or node transition has been established for each of the N nodes or states; thus, the algorithm can be applied to situations where there may be errors in the control that would necessitate a closed loop control philosophy. The algorithm is compared in both data core storage and maximum number of operations with a dynamic programing algorithm

and with an algorithm of Ford and Fulkerson. The algorithm is found to compare favorably in both core storage and maximum operations for a large number of nodes.

Langley Research Center,
National Aeronautics and Space Administration,
Hampton, Va., February 25, 1974.

REFERENCES

1. Rossi, Bruno: Optics. Addison-Wesley Pub. Co., Inc., c.1957.

2. Dijkstra, E. W.: A Note on Two Problems in Connexion With Graphs. Numer. Math., vol. 1, 1959, pp. 269-271.

3. Bellman, Richard: On a Routing Problem. Quart. Appl. Math., vol. XVI, no. 1, Apr. 1958, pp. 87-90.

4. Ford, Lester R., Jr.; and Fulkerson, D. R.: Flows in Networks. Princeton Univ. Press, 1962.